

# **Hierarchical Storage Interface (HSI)**

# Table of Contents

Preface .....	5
Introduction .....	6
How to Run HSI .....	7
Authentication Methods .....	7
Customizing the HSI Prompt .....	7
HSI Command Line .....	8
HSI Help .....	9
HSI Command Line Options .....	9
HSI Keywords .....	11
File Names .....	13
Path Names .....	13
FTP Compatibility/Differences .....	14
Operating System Considerations .....	14
Examples .....	15
Customization .....	16
Bugs and Deficiencies .....	16
HSI Command Dictionary .....	17
ADD .....	17
ADOPT .....	17
ANNOTATE .....	17
BELL .....	18
BYE .....	18
COPY .....	18
CD .....	19
CDLS .....	19
CGET .....	19
CHACCT .....	19
CHCOS .....	20
CHGRP .....	20
CHMOD .....	21
CHOWN .....	22
CLOSE .....	22
CONNECT .....	22
CP .....	23
CPUT .....	24
CRENAME .....	24
DEBUG .....	25
DELETE .....	25
DIR .....	25
DU .....	26
DUMP .....	26
ECHO .....	27
END .....	27
ERASE .....	27

EXIT .....	27
FIND .....	28
FIREWALL .....	29
FREE .....	29
GET .....	30
GLOB .....	31
GROUPS .....	31
HELP .....	31
HISTORY .....	32
ID .....	32
IDLETIME .....	33
IN .....	33
KEEP .....	33
LCD .....	33
LCDLS .....	34
LIST .....	34
LLS .....	34
LMKDIR .....	34
LN .....	35
LOG .....	35
LPWD .....	35
LS .....	36
LSCONNECTIONS .....	37
LSCOS .....	37
LSFSET .....	37
LSFILESETS .....	37
LSJUNCTIONS .....	38
LSSITES .....	38
MDELETE .....	38
MD .....	38
MDEL .....	38
MGET .....	38
MIGRATE .....	39
MKDIR .....	39
MOVE .....	39
MPUT .....	39
MV .....	40
NEWACCT .....	40
OPEN .....	41
OUT .....	41
PLOCK .....	42
PROGRESS .....	42
PROMPT .....	42
PUNL .....	43
PURGE .....	43
PUT .....	44
PWD .....	45

QUIT .....	45
R .....	45
RD .....	45
REC .....	45
REMOVE .....	45
REPLACE .....	46
RM .....	46
RMDIR .....	46
SAVE .....	46
SEND .....	47
SET .....	47
SETCON .....	49
SETDRIVE .....	49
SHARED_MEM .....	50
SHOW .....	50
SHOWCON .....	50
STAGE .....	51
STORE .....	51
SU .....	52
SUDO .....	52
SWITCH .....	53
TOUCH .....	53
UMASK .....	53
VERBOSE .....	53
VERSION .....	54
WHOAMI .....	54
Multiple HPSS Systems .....	55
Using HSI to Access Multiple HPSS Systems .....	55
Commands Used for Multi-HPSS HSI Sessions .....	57
Example Scenario for a Multi-HPSS Session .....	57
Establish Initial Connection .....	58
Establish Connection to Another Site .....	58
Logging HSI Sessions .....	60
HSI Environment Variables .....	61
Disclaimer .....	63
Keyword Index .....	64
Alphabetical List of Keywords .....	65
Date and Revisions .....	66

# Preface

**Scope:** HSI is a utility that communicates with the High Performance Storage System (HPSS) with a user-friendly interface that makes it easy to transfer files and manipulate files and directories using familiar UNIX-style commands. HSI supports recursion for most commands as well as CSH-style support for wildcard patterns and interactive command line and history mechanisms, among its many other features.

For an overview of archival storage features at LC (open and secure), including access issues and solutions to common storage problems, see the EZSTORAGE (URL: <https://computing.llnl.gov/LCdocs/ezstorage>) guide.

**Availability:** HSI runs on LC's open and secure AIX (IBM) and Linux/CHAOS production machines.

**Consultant:** For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov), SCF e-mail: [lc-hotline@pop.llnl.gov](mailto:lc-hotline@pop.llnl.gov)).

This manual was adapted from the HSI manual written by Michael Gleicher, Gleicher Enterprises, LLC. For more information on HSI AND HTAR, go to <http://www.mgleicher.us> (URL: <http://www.mgleicher.us>).

**Printing:** The print file for this document can be found at:  
OCF: <https://computing.llnl.gov/LCdocs/hsi/hsi.pdf> (URL: <https://computing.llnl.gov/LCdocs/hsi/hsi.pdf>)

# Introduction

HSI provides a Unix shell-style interface to the High Performance Storage System. Directories and files can be listed using the "ls" command, and traversing directories can be accomplished with the "cd" command.

When HSI is launched, it performs the following actions:

- Parses command line options (see [HSI Command Line Options](#) (page 17) for more details).
- Reads startup files (the user's \$HOME/.hsirc, and the system-wide hsirc file that is optionally installed by the system administrator), if they exist. In general, most settings that are defined in the system-wide hsirc file can be overridden by the user's private .hsirc file.
- Authenticates using one of the mechanisms that were enabled when the application was compiled, such as kerberos or a username/password combination. There are several ways to override the default mechanism, as described in the [HSI Authentication](#) (page 7) section.

For more information about LC's archival storage system, see the [HPSS](#) (URL: <https://computing.llnl.gov/LCdocs/HPSS>) guide.

If your file-transfer needs specifically involve placing many files into or retrieving them from a remote archive (TAR-format library) file, consult the [HTAR Reference Manual](#) (URL: <https://computing.llnl.gov/LCdocs/htar>) for another LC-designed, locally deployed tool tailored to efficiently managing large archives in storage or on any preauthenticated FTP server.

# How to Run HSI

## Authentication Methods

HSI supports several different methods by which a user can authenticate his or her self. The methods that are enabled at a site are defined by the HPSS administrator when the HSI package is compiled.

The `-A` command line option can be used to specify the authentication mechanism to be used, for example:

```
hsi -A kerberos
```

The following authentication methods are available:

- **combo (previously known as "dce"):** For this method, a user name and password combination are specified. For backward compatibility, the mechanism can still be specified as "dce", however, this is deprecated and will be removed in a future release.
- **keytab:** For this method, a kerberos keytab file or a unix keytab file is read by the HSI library and passed to the HSI Gateway Server over the link after first encrypting the contents. On the server side, the file is decrypted and verified.
- **kerberos:** This mechanism uses the MIT Kerberos libraries to verify the user's existing credentials. A kerberos keytab file can also be specified for kerberos authentication. If so, the "kinit" program is first used to obtain the user credentials, and then authentication proceeds normally.
- **ident:** This mechanism uses the trusted server protocol to obtain the user's identity.
- **gsi:** This mechanism uses the Globus GSI protocols to obtain the user's identity.

## Customizing the HSI Prompt

The default HSI prompt string ("`?` " for the initial line, and "`>` " for continuation lines) may be customized by setting the "PS1" and "PS2" keywords, or by specifying the PS1 and PS2 settings in the global or private `hsirc` files.

For example, setting the strings as follows:

```
PS1 = "%d[%H]%w3->" PS2 = "[%H]continue: "
```

would produce a prompt string similar to this for the initial prompt:

```
R:[hcdp01]/users/sdsc/gleicher->
```

and a prompt similar to this for continuation lines:

```
[hcdp01]continue:
```

Within a prompt string, characters are copied literally, except for substitutable parameters of the form `%x`, where `x` is one of the following:

- |   |   |
|---|---|
| C | DCE Cellname (currently recognized, but unused) |
| c | current connection ID                           |
| d | current logical drive ID in the form A:         |
| D | system date in yyyy/mm/dd format                |

h	canonical hostname for current connection
H	remote HPSS hostname for current connection
I	remote HPSS IP address for current connection
L	current login principal
s	site name
N	current command number (1:N)
T	system time in hh:mm:ss format
W[n]	current HPSS working directory for connection, with an optional number of directory paths to include (from right to left), as specified by the optional numeral following the %W
w[n]	current HPSS working directory, with "." substitution in long directory paths, with optional number of directory paths (from left to right) in the current directory path to include if "." substitution is required.
%	% character

## HSI Command Line

The HSI Command Line has the following format:

```
hsi [options] [command [;command [;...]] ]
```

If an optional command-string is specified, then HSI is running in single execute line mode (also known as "one-liner" mode). In this mode, HSI will execute the command-string and then terminate. Multiple commands may be specified and separated by the semicolon (;) character. The command line may need to be enclosed in single or double quotes to protect it from expansion by the shell program that launches HSI.

You can also use HSI interactively by entering the command HSI without any arguments. Then simply enter HSI requests in response to the question-mark (?) prompt (or whatever the prompt string has been set to). Terminate the HSI session with the END command or one of its several aliases, such as BYE or QUIT.

Single-line execution is often used when HSI is run from within UNIX scripts and pipelines. An example of using HSI with two commands on the execute line might be:

```
hsi "cd /users/project1;ls -l *.c"
```

See [Operating System Considerations](#) (page 14) for more information on using HSI in scripts.



# HSI Help

Within an HSI session, help information for a command may be obtained by either entering a command with no parameters, e.g., PUT, or for commands that do not require parameters, enter the command name followed by

-?

For example,

**ls -?**

The HELP command (page 31) will display a short summary of available HSI commands.

## HSI Command Line Options

The basic HSI command options are briefly described below. For complete details on all HSI commands, see the HSI Command Dictionary (page 17).

**-help | -?** Displays a built-in synopsis of HSI usage. If used as the only command line option, HSI exits instead of entering interactive mode.

**-a acctID | acctName**

Specifies the account ID or account name to be used for creating new files and directories.

**-A auth\_mechanism**

Specifies the authentication mechanism to be used when HSI starts up. Only authentication mechanisms that are available on both the client and HSI Gateway Server can be used successfully. Valid values for auth\_mechanism are:

**combo** User name and password (or SecurID token if RSA SecurID tokens are in effect).

**Note:** the deprecated "dce" authentication mechanism is still recognized as an alias for the "combo" mechanism, but its use should be discontinued, as it will be removed in a future release.

**kerberos** Specifies MIT kerberos.

**keytab** A UNIX or Kerberos-style keytab.

**gsi** The Globus GSI certificate protocol.

**local** Protocol used for sites that allow users to authenticate using their AIX (or (AIX only) LDAP) passwords on the HSI Gateway Server machines.

**-c cred\_file** Specifies the credentials cache file to be used if Kerberos authentication is used.

**-d *debug\_level*** Specifies the debugging level to enable when HSI starts up. *debug\_level* is an integer value between 0 and 4. Level 0 (the default) specifies no debugging, levels 1-4 specify successively higher debug levels.

In general, HSI debugging output is useful only to the developers for troubleshooting problems.

**-e** "echo" flag. If enabled on the command line or by the ECHO command, HSI will display command lines that are read from IN files.

**-E *editor-style*** Specifies the editing style used when editing command lines. *editor-style* is either vi or emacs. The default, if not specified in the .editrc file, is vi.

**Note:** This option is effective only if HSI was built with line-editing enabled.

**-G *globus\_grid\_proxy\_path***

If Globus GSI authentication (for example, if **-A gsi** is specified), this option can be used to specify an alternative path to the proxy created by the *grid-proxy-init* command.

**-h *hostname***

Directs HSI to attempt to connect to *hostname*, which can be either a name or an IP address.

**-k *keytab\_path*** Specifies the path to the keytab file if keytab authentication is used.

**Note:** The keytab file will not be used by HSI if it has other than owner permissions.

**-l *login\_name*** Specifies the user's login name. This option overrides any login name specified in the *hsirc* file(s). If no login name is specified, the default is to use the local login name on the client machine on which HSI is being run.

**-O *pathname*** Used by applications that run HSI via the "system" or "popen" calls. It causes HSI to write all output that would normally be written to the terminal or stderr to the file specified by *pathname*. The option takes effect immediately when it is encountered and normally is the first option on the command line (if used) in order to capture any messages related to parsing the command line. It also causes the "quiet" flag to be set (disabling extraneous messages from being displayed, such as the message-of-the-day), and the "verbose" command response flag to be disabled. In addition, it causes all interactive file-transfer progress messages to be disabled.

**-o** Overrides the <no-login> flag, if set by the HPSS administrator. This option can only be used by root. It is deprecated and will be removed in a future release.

**-P** popen command mode. This option is similar to the **-O** option, but causes all output to be directed to stdout, which is normally redirected to a process that starts HSI with the **popen(3)** system call. It also results in setting quiet (no extraneous messages) mode, disabling verbose response message, and disabling interactive file transfer messages.

- p *port*            Specifies the port number to be used when connecting to the HSI Gateway Server. This option is only effective if the -h *hostname* option is specified.
- q                    Enables "quiet" mode. In this mode, extraneous messages, such as the interactive file transfer progress messages, as well as the startup "Message of the Day" are not displayed.
- s *site\_name*       Specifies the global or private hsirc stanza name to be used. *site\_name* must match the name of one of the stanzas in either the global hsirc file or the user's private \$HOME/.hsirc file.
- V                    Prints HSI version information without requiring a connection to the HSI Gateway Server. If used as the only command-line option, HSI exits instead of entering interactive mode.
- v                    Enables verbose mode for command output. Verbose mode is used for providing interactive feedback, such as displaying markers on the listable output file when performing operations such as PUTS or GETS, or searching for files via the FIND and DU commands.

## HSI Keywords

The following keywords are available in almost every HSI command. They are set by one of the following mechanisms:

1. Setting a value for the session command

**set *keyword=value***

**Note:** The "SET" command string can optionally be omitted, as HSI assumes that any command of the form *keyword = value* is a "SET" command.

2. Setting a value for the current command

***command keyword=value***

The following keywords, and their minimum abbreviation, are listed below.

**ba[ckup]=on | off**

automatically renames existing file on get/put. Default is on.

**col[umns]= *numeric value***

Specifies number of columns per line. This value is used by the **ls** command when determining the number of entries per line to list. It is changed whenever the screen size changes.

**copies=n**

number of copies to store. The default is set by the HPSS administrator in the server configuration file.

**cos[id]=auto | dualcopy | nodualcopy | *cosid***

class of services to use. Default is "auto", which selects class of service for you. You should only change this if you need to use a different class of service, for example, use

"dualcopy" to go to two copies on tape for mission critical data. Use the `lscos` command to see the classes of service defined for the site; be aware that these are highly site-dependent. **Note:** The **dualcopy** and **nodualcopy** settings are LLNL-specific options.

`coslist= name`

"named Class of Service" to use. Default is no named COS. Named Classes of Services are groups of COSs that are defined by the HPSS administrator. Use "`lscos -n`" to display a list of named COSs for the current active connection.

`dcreate[mode]= octal_value`

Octal mode to use when creating directories. This value is defined in the global or private `hsirc` file's **dcreate\_mode** setting, or to 0777 by default. The current `umask` value is used in conjunction with the mode setting in order to yield the effective permissions that are set.

`dirn= pathname`

(`n` is omitted (meaning `dir0`) or 0 to 9) (The default pathname is your HPSS home directory) `dir0` or `dir` assigns the current working directory. `dirn` sets up a shorthand name for the `n`th working directory as `pathname`. This directory can then be used in commands by using the "tic" notion. For example `ls 3`

`family[id]= numeric_value`

Set default file family ID to be used when creating new files. The default is either 0, or the value defined in the global or private `hsirc` files for your site.

`fcreate[mode]= octal_value`

Octal mode to use when creating files. This value is defined in the global or private `hsirc` file's **fcreate\_mode** setting, or to 0777 by default. The current `umask` value is used in conjunction with the mode setting in order to yield the effective permissions that are set.

`iob[ufsize]= numeric_value`

I/O buffer size to use when transferring files. The default is set by the HPSS administrator when HSI is built, normally 8MB. The `numeric_value` can optionally be followed by any of the following multipliers, with no intervening space: "k", "kb", "m", "mb", "g", "gb", "t", "tb", "p", "pb" for kilobytes/megabytes/gigabytes/terabytes/petabytes. The maximum buffersize is normally 32MB, but is defined at build time by the HPSS administrator.

`lines= numeric_value`

Sets the number of lines per page for the terminal. This value was used for a curses-based help package for a previous version of HSI, but is currently unused.

`promptl[en]= numeric_value`

Sets the prompt length in characters

promptd[irlen]= *numeric\_value*

Sets the maximum number of characters to use when expanding the %w metacharacter in the command prompt. This is the number of directory components to include from the righthand side of the pathname if the prompt must be truncated to *promptlen* characters.

PS1= *string*

Specifies the prompt string to use for initial command prompting. See the [Customizing the HSI Prompt section](#) (page 7) for details on the options for *string*. The default setting for this value is "?" or the value of the PS1 setting in the global or private hsirc file.

PS2= *string*

Specifies the prompt string to use for continuation lines. See the [Customizing the HSI Prompt section](#) (page 7) for details on the options for *string*. The default setting for this value is ">" or the value of the PS2 setting in the global or private hsirc file.

pwid[th]= *auto / numeric\_value*

Sets the parallel stripe width to use when transferring files. If **pwid** is set to "auto" (the normal case), HSI automatically determines the optimum stripe width to use.

You can set these keywords for individual commands, for your session, or in saved keysets.

## File Names

HPSS file names and UNIX system file names are both mixed case, and case **is** important. The maximum value for *pathname* (including slashes and wildcards) is 1024 characters. The maximum value for a file or directory name (except wildcard path) is 256 characters.

## Path Names

Valid characters for a path parameter are:

a-z A-Z 0-9 \$ % + - . \_ #

Wildcard characters are

\* ? ^ [ ] { }

They may be used in all commands that reference either local or HPSS paths, including the local : HPSS form. Note that names containing wildcard characters must resolve to a single node for some commands, such as SET DIR *n* =*pathname*, or MV *path1 path2... npath*, in which *npath* must resolve to a single directory node. UNIX-style pathname prefixes ~, ., and.. are also allowed on all file name/HPSS-path name references.

## FTP Compatibility/Differences

HSI supports several of the commonly used FTP commands, including DIR, GET, LS, MDELETE, MGET, PUT, MPUT, and PROMPT, with the following differences:

- The DIR command is an alias for LS in HSI. The LS command supports an extensive set of options for displaying files, including wildcard pattern-matching and the ability to recursively list a directory tree.
- The PUT and GET family of commands support recursion.
- There are "conditional put" and "conditional" get commands (CPUT, CGET).
- The syntax for renaming local files when storing files to HPSS or retrieving files from HPSS is different than FTP. With HSI, the syntax is always LOCAL\_FILE : HPSS\_FILE, and multiple such pairs may be specified on a single command line. With FTP, the local filename is specified first on a PUT command, and second on a GET command.

For example, when using HSI to store the local file "file1" as HPSS file "hpss\_file1" and then retrieve it back to the local file system as "file1.bak", the following commands could be used:

```
put file1 hpss_file
get file1.bak : hpss_file1
```

With FTP, the following commands could be used:

```
put file1 hpss_file1
get hpss_file1 file1.bak
```

- The "m" prefix is not needed for HSI commands; all commands that work with files accept multiple files on the command line. The "m" series of commands are intended to provide a measure of compatibility for FTP users.

## Operating System Considerations

There are a few issues to remember when building UNIX scripts to communicate with HPSS.

- HSI LS outputs a single entry per line and no heading information when "-1" (digit 1, not to be confused with lowercase letter L) is specified.
- Do not use wildcard paths in scripts unless you protect them from the shell with ' or "'. You're probably better off using IN-files if you can, because they are read directly by HSI without having to first be passed through the shell.
- ALL output is written to standard error unless an OUT command is in effect.

Using "-" as the local file name specifies standard input (or output). When this option is used, the maximum-sized file ( $2^{64}$ ) is assumed; less than this amount may be sent (by closing the input pipe).

```
tar cf -file1 file2 | hsi save - : project.tar
```

In the example above (and any time the data is transferred into HPSS via a UNIX pipe), HSI cannot determine the file size ahead of time. In this case, if *cos=auto* is in effect (the default), the file may have to be recopied within HPSS to move it to the correct storage class. If you know ahead of time which class of

service (COS) is the correct one for a piped file, you can assist HPSS by requesting the appropriate COS on the command.

In the example above, if it is known that BigFile belongs to COS 20 (i.e., the HPSS administrator has created a COS whose ID is 20 to contain the files above), the command line could be rewritten as:

```
tar cf - file1 file2 | hsi save cos=20 - : project.tar
```

This forces the file(s) to be written to COS 20. In this case, HPSS will not have to recopy the file in order to place it on the correct storage devices.

On UNIX systems, the HSI utility will always wait for a reply from the HPSS server when HSI is first invoked interactively or when a one-liner is executed. If HPSS is unreachable and you choose not to wait, you can break out of the utility with CTRL-C.

## Examples

Save a "tar file" of C source programs and header files:

```
tar cf - *.*[ch] | hsi put - : source.tar
```

**Note:** the ":" operator which separates the local and HPSS pathnames must be surrounded by whitespace (one or more space characters)

Restore the tar file source kept above and extract all files:

```
hsi get - : source.tar | tar xf -
```

Get all files in the subdirectory subdira which begin with the letters "b" or "c" (surrounding the wild-card path in single quotes prevents shells on UNIX systems from processing the wild card pattern):

```
hsi get 'subdira/[bc]*'
```

Save your local files that begin with the letter "c" (let the UN\*X shell resolve the wild-card path pattern in terms of your local files by not enclosing it in quotes):

```
hsi put c*
```

Delete all files beginning with "m" and ending with 9101 (note that this is an interactive request, not a one-liner request, so the wildcard path does not need quotes to preserve it):

```
hsi <RETURN>  
? delete m*9101
```

Interactively delete all files beginning with H and ending with a digit, and ask for verification before deleting each such file.

```
hsi <RETURN>  
? mdel H*[0-9]
```

Interactively descend into the "Source" directory and move all files which end in ".h" into a sibling directory (ie, a directory at the same level in the tree as "Source") named "Include":

```
hsi <RETURN>  
? cd Source  
? mv *.h ../Include
```

## Customization

Each user may have a file called `.hsirc` in their home directory, in order to specify the HPSS Gateway machine's hostname or IP address, or to specify other options such as the prompt string, whether to enable or disable automatic backups when storing or retrieving files that already exist, and other options. The `hsirc` file is documented on the [HSI web site](http://www.mgleicher.us) (URL: <http://www.mgleicher.us>).

## Bugs and Deficiencies

- HSI should not blindly interpret embedded "=" signs as "keyword=value", as this makes it necessary to backslash-quote any pathname that contains an equals sign.
  - HSI does not expand wildcards for the local pathname(s) for the LLS and LCDLS commands
- Interactive help should be available for individual commands



# HSI Command Dictionary

Aliases for commands, if any are shown in parantheses following the "real" hsi command.

Commands may be abbreviated; the minimum abbreviation for each command is shown by the characters preceding the "[" (if any) for the command name.

## ADD

See MKDIR (page 39).

## ADOPT

SYNTAX:

```
ado[pt] [-l] [-h] [keyset]
```

ROLE:

Reads an HPSS-resident or local filesystem-resident keyset file.

OPTIONS:

- l               adopt local keyset.
- h               adopt HPSS-resident keyset.

## ANNOTATE

SYNTAX:

```
anno[tate] [-R] [-e] [-d] [-f] [-A "string"] path...
```

ROLE:

Adds annotation text to existing files or directories. The annotation string may be a maximum of 255 characters. It is silently truncated during expansion if it exceeds this length.

The annotation string may contain backslash-escaped characters which are replaced with their normal ASCII equivalents as follows:

- \a alert -> bell character
- \n newline
- \f formfeed
- \r carriage return
- \t horizontal tab
- \v vertical tab
- \\ backslash
- \? question mark
- \' single quote
- \" double quote

Note: Two consecutive backslash characters are required in order to enter one of the above, since the HSI parser treats the first backslash as a "quote" character while initially scanning the command. The second backslash and the character following are then evaluated by the annotate command.

OPTIONS:

- R                recursively traverse directories in the specified path(s)
- e                erase any existing annotation
- d                operate only on directory objects
- f                operate only on file objects
- A                annotation string (including metacharacters). An empty string is treated as if the -e option was specified.

## BELL

SYNTAX:

**bell**

ROLE:

Toggle sounding a bell when keyboard input is needed.

## BYE

See END (page 27).

## COPY

SYNTAX:

**c**

See also CP (page 23).

## CD

### SYNTAX:

**cd** [0..9] [drive:] *path*

### ROLE:

Changes current HPSS Working Directory to *dirname*. The default is your HPSS home directory. Working directory 0 is used if the directory number is not specified. Working directory number. 0 is used if the directory number is not specified.

### OPTIONS:

-? Prints help for the CD command. CD with no parameters is semantically equal to CD0 ~. The following are exactly equivalent: CD0 or CD - or CD ~

## CDLS

### SYNTAX:

**cdl** [s] [options] [*path...*]

### ROLE:

Change the current directory to *pathname* and list its contents using the listing options specified by *options*. See the [LS command](#) (page 36) for listing options.

## CGET

### SYNTAX:

**cget** [ *options* ] [ *localfile* : ] *hpsspath...*

### ROLE:

Conditionally copy an HPSS file *hpsspath* to *localfile* if it does not exist locally. Default *localfile* is *hpsspath*.

See [GET](#) (page 30) for a description of all of the command options.

## CHACCT

### SYNTAX:

**chac[ct]** [-R] *new-acct path...*

### ROLE:

Change account ID for HPSS file path to *new-acct*.

### OPTIONS:

-R recursion flag

## CHCOS

### SYNTAX:

```
chcos [-h] [-N] [-R] new-cos | auto | dualcopy | nodualcopy path...
```

### ROLE:

Change Class of Service for path to new-cos

### OPTIONS:

- h Follow symbolic links during recursion. The default behavior is not to follow symbolically-linked directories.
- N disables auto-scheduling of files (by Volume and Position). The normal default is to organize the list of files by Volume ID and position, in order to optimize tape mounts
- R recursively traverse directories in the specified path(s)
- Z internal autoscheduler debug flag. Displays scheduling results but doesn't issue the HPSS chcos call(s)
  
- new\_cos* specifies the COS ID to be used, which must be one which you are eligible to use for the HPSS connection(s) specified by the "path" parameter(s) (use LSCOS to see a list of COS IDs).

COS ID 0 ("chcos 0") is always permitted; it removes files from the HPSS chcos queue.

Use CHCOS AUTO to use auto-COS selection to choose the new COS ID, based upon current settings for # copies, account ID, group ID, and user ID

Use CHCOS DUALCOPY to use the dual copy COS.

Use CHCOS NODUALCOPY to return to the normal auto-COS selection. (This is an LLNL-specific command)

## CHGRP

### SYNTAX:

```
chgrp [-h] [-R] group-name path...
```

### ROLE:

Change group owner for path to group-name.

### OPTIONS:

- h For symlinks, change ownership of symlink instead of the object which it references.  
**Note:** HPSS does not provide this capability, so symlinks are silently ignored if this option is specified].
- R recursion flag

# CHMOD

## SYNTAX:

```
chmod [-d] [-f] [-h] [-H] [-R] mode path...
```

## ROLE:

Change permissions for HPSS files.

## OPTIONS:

- d changes apply only to directory objects
- f changes apply only to file objects
- h suppresses a mode change for a file or directory pointed to by a symbolic link
- H If the -R option is specified, symbolic links on the command line are followed. Default is not to follow symlinks for any path specified on the command line. Symbolic links encountered during recursion are never followed.
- R descends directories recursively, changing the file mode bits of each directory and of all files matching the specified pattern, taking into account the other flags. If a symbolic link is encountered, and the link points to a directory, the file mode bits of that directory are changed, but the directory is not further traversed.

The mode of each named file is changed according to mode, which may be absolute or symbolic. An absolute mode is an octal number constructed from the logical OR of the following modes:

0400 read by owner

0200 write by owner

0100 execute (search in directory) by owner

0070 read, write, execute (search) by group

0007 read, write, execute (search) by others

A symbolic mode has the form:

```
[who] op permission [op permission]...
```

The *who* part is a combination of the letters *u* (for user's permissions), *g* (group) and *o* (other). The letter *a* stands for *all*, or *ugo*. If *who* is omitted, the default is *a* but the setting of the file creation mask (see `umask(2)`) is taken into account.

*op* can be + to add permission to the file's mode, - to take away permission and = to assign permission absolutely (all other bits will be reset).

*permission* is any combination of the letters *r* (read), *w* (write), *x* (execute), *X* (set execute only if file is a directory or some other execute bit is set). Letters *u*, *g*, or *o* indicate that permission is to be taken from the current mode. Unlike Un\*x systems, a value must be specified with = to remove all permissions, specify either absolute or *-rwx*.

When the -R option is given, chmod recursively descends its directory arguments, setting the mode for each file as described above. When symbolic links are encountered, their mode is not changed and they are not traversed.

## CHOWN

### SYNTAX:

```
chown [-h] [-R] owner-name[:group-name] path...
```

### ROLE:

Changes owner of *path* to *owner-name*

### OPTIONS:

- h suppresses a change for a file or directory pointed to by a symbolic link. Note: this option, if specified, currently causes symbolic links to be ignored with a warning message, as HPSS does not provide the capability to change the ownership of a symbolic link.
- R Recursion flag

## CLOSE

### SYNTAX:

```
close [connectionID|drive:...]
```

### ROLE:

Close a connection to an HPSS server

### OPTIONS:

- connection ID* numeric connection id
- drive* logical drive prefix (including colon ":" character)

Use LSC[ONNECTIONS] for list of current open connections. If no connections are specified, the current connection is closed. It is an error to attempt to close the last active connection.

## CONNECT

See OPEN (page 41).

## CP

### SYNTAX:

```
cp | c[opy] [ options ] sourcefile sinkfile
```

```
cp | c[opy] [options] source... sinkDirectory
```

### ROLE:

Copies one file to another or copies files into a directory

### OPTIONS:

- B create backup of target file by renaming if it exists (same as "backup=on")
- C cache purge option. If specified, purge source files from HPSS disk cache after a successful copy. Normally used only when it's expected that files will only be read once (or infrequently), to help optimize HPSS disk cache use.
- f force removal of the target file instead of renaming (same as "backup=off")
- h copy symbolic links (default is to read through symlinks)
- i prompt before copying files (interactive mode)
- m interhpss copy method ("local" or "server"). Default is "server"
- n only copy files modified within the specified number of days
- p preserve timestamp
- R or -r recursively copy SourceDirectories
- S disable staging of the source file (attempts to read directly from tape). This option may be useful for large files that will only be retrieved once (or infrequently) to avoid the time and space that it takes to stage the file(s) onto HPSS disk cache.
- x use extended I/O calls. The "which" parameter is "source", "sink", or "both". This option is normally only used when "firewall" mode is in effect, and designates whether firewall-type I/O should be used when reading ("source") files, writing files ("sink") or both ("both"). One scenario where this is useful is copying files between HPSS systems, when one or the other HPSS systems lives behind a firewall.

## CPUT

### SYNTAX:

```
cput [options] [localfile : ] hpsspath...
```

### ROLE:

Conditionally save or replace HPSS file *hpsspath* with *localfile* if it has been updated.

See the PUT (page 44) command description for a description of *options*

## CRENAME

### SYNTAX:

```
cren[ame] [-d] [-ocos id[,id...]] [-ncos id[,id...]] [-s]  
new_path orig_path
```

### ROLE:

Conditionally rename and optionally delete files.

### OPTIONS:

- d delete original file if the rename operation succeeds
- s suffix string to append to orig\_path when renaming it. Default is "~"
- ocos Filter option. If specified, the Class of Service of file "orig\_path" must be one of the comma-separated COS IDs specified, or the command will fail.
- ncos Filter option. If specified, the Class of Service of file "new\_path" must be one of the comma-separated COS IDs specified, or the command will fail.



## DEBUG

### SYNTAX:

**debug** *level* |on|off

### ROLE:

Set debugging level.

### OPTIONS:

level	a number in the range 0 to 5. 0 disables debug messages, levels 1-5 enable increasingly higher levels of debug, roughly corresponding to the following
1	event and high-level information messages
2	normal debug messages that help track the flow of command processing
3	higher level of debug used for inner loops, table traversals, etc.
4	I/O debug - mover message traces
5	trace-level debug
on	equivalent to "debug 1"
off	equivalent to "debug 0"

## DELETE

See RM (page 46).

## DIR

See LS (page 36).

## DU

### SYNTAX:

```
du [ -p ] [ -s ] [ -a ] [-b date ] [-e date ] [-s] [-w n] [ path... ]
```

### ROLE:

Summarize HPSS file space usage in specified path or current directory

### OPTIONS:

- du** gives the number of kilobytes contained in all files and, recursively, directories within each specified directory or file name. If name is missing, '.' is used.
- a** causes an entry to be generated for each file. If neither **-s** or **-a** is specified, an entry is generated for each directory only.
- b** *date* only counts files written since the specified date. *date* is of the form yyyy/mm/dd.
- e** *date* only counts files written before or on the specified date. *date* is of the form yyyy/mm/dd. Note that the **-w** option, if specified, must reference a date prior to or equal to the one specified by the **-e** option or an error will occur.
- k** print values in kilobytes (1k=1024). Default is 512 byte blocks
- s** causes only the grand total to be printed.
- w** *n* only counts files written within the last '*n*' days

## DUMP

### SYNTAX:

```
dump [-R] [-x] path...
```

### ROLE:

List metadata information for HPSS Nameserver objects.

### OPTIONS:

- R** recursively traverse directories in the specified path(s)
- x** dump extended attributes for object

## **ECHO**

SYNTAX:

**echo**

ROLE:

Toggle for displaying command input from IN files

## **END**

SYNTAX:

**ex[it]**

**q[uit]**

**bye**

ROLE:

Terminate HSI execution

## **ERASE**

See RM (page 46).

## **EXIT**

See END (page 27).

# FIND

## SYNTAX:

**find** *pathname-list expression*

## ROLE:

Find nodes in the directory structure matching specified criteria. (The implementation is a subset of the Un\*x version of find(1), but the syntax is almost identical.

The *find* command recursively descends the directory hierarchy for each pathname in the *pathname-list*, seeking files that match a boolean (logical) expression written in the primaries given below. In the descriptions, the argument *n* is used as a decimal integer where *+ n* means more than *n*, *-n* means less than *n*, and *n* means exactly *n*.

*-name filename*

True if the *filename* argument matches the current file name. Wildcard characters are allowed, with or surrounding quotes ("" or ``).

*-perm onum*

True if the file permission flags exactly match the octal number *onum*. Only the lowest 3 octal digits are significant.

*-cos*

*cosid* True only for plain files whose Class of Service ID matches that specified by *cosid*

*-prune*

Always yields true. Has the side effect of pruning the search tree at the file. That is, if the current path name is a directory, find will not descend into that directory.

*-type*

True if the type of the file is *c*, where *c* is one of:

d for directory

f for plain file

l for symbolic link

*-user*

True if the file belongs to the user *uname*. If *uname* is numerical, it is taken as a user ID.

*-nouser*

True if the file belongs to a user not in the passwd database.

*-group*

True if the file belongs to group *gname*. If *gname* is numeric, it is taken as a group ID.

*-nogroup*

True if the file belongs to a group not in the group database.

*-size [+ | -] n*

If *+* is specified, true if the filesize is *> n* bytes. If *-* is specified, true if the filesize is *< n* bytes. Otherwise, true if the filesize is exactly *n* bytes.

*-atime n*

True if the file has been accessed in *n* days. Note that the access time of directories in *pathname-list* is changed by the find command.

*-mtime n*

True if the file has been modified in *n* days.

- ctime *n* True if the file has been changed in *n* days. "Changed" means that the file's "st\_mtime" field has changed.
- print Always true; the current pathname is printed.
- ls Always true; causes current pathname to be printed together with its associated statistics.
- newer *filename* True if the current file has been modified more recently than the HPSS file *filename*.
- depth Always true; causes descent of the directory hierarchy to be done so that all entries in a directory are acted on before the directory itself.
- exec command (DEFERRED - intended for use 'get/put', etc)  
True if the executed command returns a zero value as exit status. The end of command must be punctuated by an escaped semicolon. A command argument { } is replaced by the current pathname.
- ok command (DEFERRED)  
Like -exec except that the generated command is written on the standard output, then the standard input is read and the command executed only upon response y.

## FIREWALL

### SYNTAX:

**fire[wall]** [-on | -off] [-c] [-s] [ *drive...* ]

### ROLE:

Enable or disable firewall mode for a connection. When firewall mode is enabled, store-and-forward I/O is used for file transfers instead of the normal mode where HPSS movers connect to HSI. In firewall mode, auto-scheduling is disabled for the GET command family.

## FREE

### SYNTAX:

**fr[ee]** [-h] [-l] *keyset*

### ROLE:

Delete the kept set of keyword values, *keyset*.

### OPTIONS:

- l deletes local keysets.
- h (default) deletes HPSS-resident keysets.

# GET

## SYNTAX:

```
g[et] [ options ] [ localfile : ] hpsspath... [ <<MARKER ]
```

## ROLE:

(mget, rec[v]) Copy a HPSS file *hpsspath* to *localfile*. Default *localfile* is *hpsspath*.

**Note:** MGET prompts for "yes/no" in interactive mode. Use PROMPT to toggle interactive mode.

## OPTIONS:

- A enable auto-scheduling for retrievals in order to optimize tape mounts
- B | -b backup option. Renames any existing local file by appending "~"
- C purge source file from disk cache after file(s) are copied successfully
- F on | off enable or disable the use of HPSS Local File Mover I/O, overriding any configuration option(s)
- h symlink option. If specified, create local symlink if HPSS symlink. If not set, reads through HPSS symlinks when copying files to local filesystem
- k keep partially transferred file(s) if error(s) are encountered
- L path specifies pathname to a file containing a list of file segments to be retrieved.
- N disable auto-scheduling of retrievals
- O tuple partial file transfer specification. See the usage notes below for the format of the "tuple" parameter. Multiple -O options can be specified for a command. The format of a tuple string is:  
  
**sourceOffset:sinkOffset:length**  
  
All of these numeric values may contain case-insensitive multiplier suffix strings (with no intervening space) of the form "K" or "KB" (kilobytes), "M" or "MB" (megabytes), "G" or "GB" (gigabytes), "T" or "TB" (terabytes), "P" or "PB" (petabytes). Either or both offsets may be omitted, and the length may also be omitted. Missing offsets are interpreted as meaning the current offset, and a missing length is interpreted as meaning the remainder (or all) of the file.
- p preserve timestamp. Attempts to copy the HPSS file's timestamp to the local file.
- Q if running as local root user, attempts to preserve the HPSS owner and group on the local file(s).
- R recursively copy directories
- S disable staging of the HPSS file(s); read directly from tape if file is not already in HPSS disk cache.

-T on   off	enable or disable the use of the HPSS Transfer Agent for I/O, overriding any configuration option.
-t	retransmit request. Appends to the local file, using the local file's size as the starting offset within the HPSS file.
-U   -u	update option. Only copy HPSS file to the local file if HPSS file timestamp is newer.
MARKER	provides sh-style "here"-document syntax for specifying filenames. Lines following the MARKER, which must be the last token on the line, contain pathname (which may include wildcards). The list is terminated by a line containing MARKER as the first token on the line.

**Note:** If -O,-L is specified, offsets apply only to the first file transferred.

## GLOB

SYNTAX:

**glob**

ROLE:

Toggle enabling/disabling wildcard pattern-matching for filenames. The normal mode is "enabled".

If globbing is disabled, all pathnames that are specified must exactly match, and wildcard characters \* ? { } [ ] are treated as normal characters in pathnames.

## GROUPS

SYNTAX:

**gro[ups]**

ROLE:

List groups associated with the current login context.

## HELP

SYNTAX:

**help**

ROLE:

Displays brief "help" information.

# HISTORY

## SYNTAX:

```
hist[ory] [-e ename ] [-nl] [first [last]]
```

## ROLE:

Display or edit command history

or

```
hist[ory] -s [ old=new ] [ cmd_string ]
```

## OPTIONS:

- e specifies editor name to use with first form of the command. If not specified, then the environment variables "HISTEDIT" or "FCEDIT" are checked, and if neither is set, then "/bin/edit" is used.
- l specifies <listing> action to display the commands on the listing file
- n if specified, suppresses command numbers when editing
- Aliases **r** - alias for "**hist -s**"

# ID

## SYNTAX:

```
id [-u | -G | -g [-n]] [user]
```

## ROLE:

Display current login identity information

## OPTIONS:

- G if specified, causes group list for current ID to be displayed
- g if specified, causes group ID to be displayed
- u if specified, causes user ID to be displayed
- n if specified, causes -g or -u option to display names instead of numbers

If [user] is specified, causes info to be displayed for the specified user. If not specified, info for the current login or SU identity is displayed.



## IDLETIME

### SYNTAX:

**idle[time]** [ *value* ]

### ROLE:

Set interactive session timeout value

*value* : optional idle timeout value, in seconds. If value is zero or negative, e.g. **timeout -1**, then an infinite timeout is used. If value is not specified, the current idle timeout setting is displayed.

The default value for this setting is 1800 seconds (1/2 hour).

## IN

### SYNTAX:

**in** *localfilepath*

### ROLE:

Read hsi commands from local file *localfilepath*.

The **IN** command is not allowed from within an IN file.

## KEEP

### SYNTAX:

**k[eeep]** [-h] [-l] *keyset*

### OPTIONS:

-l saves local keysets.

-h (default) saves HPSS-resident keysets.

## LCD

### SYNTAX:

**lcd** *localpath*

### ROLE:

Change local working directory to *localpath*.

**Note:** Wildcard characters in *pathname* are currently not expanded.

## LCDLS

### SYNTAX:

```
lcdl[s] [ options ] [ path ]
```

### ROLE:

Change local working directory to *path* and list the contents.

### OPTIONS:

Local list options. These may differ, depending on the local host's version of UNIX.

**Note:** Wildcard characters in *pathname* are currently not expanded.

## LIST

See LS (page 36).

## LLS

### SYNTAX:

```
lls [ options ] [ path ]
```

### ROLE:

List local (worker) directory *localpath*. Default is local current working directory.

### OPTIONS:

Local list options. These may differ, depending on the local host's version of UNIX.

**Note:** Wildcard characters in *localpath* are currently not expanded.

## LMKDIR

### SYNTAX:

```
lmd[ir] [-p] subdirname...
```

### ROLE:

(lmd) Make a new subdirectory done called *subdirname* on local filesystem.

-p makes any missing intermediate subdirectories in the *subdirname*.

## LN

### SYNTAX:

```
ln [-f ] [- h] [-n ] [ -s ] sourcename] [targetname]
```

```
ln [-f ] [- h] [-n ] [ -s ] sourcename1 [ sourcename2... ] targetdirectory
```

### ROLE:

Create symbolic link for HPSS file *sourcename*

### OPTIONS:

- f Unlink any already existing file, permitting the link to occur
- h If the target\_file or target\_dir is a symbolic link, do not follow it. This is most useful with the -f option, to replace a symlink which may point to a directory.
- n Same as -h, for compatibility with other ln implementations.
- s Create a symbolic link.

## LOG

### SYNTAX:

```
log [[>[!]] file ] [>> file ] [-]
```

### ROLE:

Write HPSS log output to *file*.

- log - close logfile if open, revert back to <no logfile> mode
- log file write HSI loggable output to <file>
- log > file same as "log file"
- log >! file same as "log file"
- log >> file append HSI response output to <file>

## LPWD

### SYNTAX:

```
lpwd
```

### ROLE:

List local current working directory

# LS

## SYNTAX:

**l**[**s**] [options] [ *path* ]

## ROLE:

(list) Display information about a directory or file *path*.

## OPTIONS:

- a list all entries, including "hidden" files
- c use time of last modification for sorting (deferred)
- d if file is a dir, list its name instead of its contents
- l (letter "l") long list format
- p put a slash after each name if the file is a directory
- r reverse alpha or age sort order, as appropriate (deferred)
- s display size as well as name if -l (numeral 1) option used
- u use time of last access for sorting instead of last modification (deferred)
- x multicolumn output format, with entries sorted across page
- A display annotation string
- C multicolumn output format, with entries sorted down the columns
- F puts a / after directory filenames, and an \* after the name if it is executable (deferred)
- H print headings on long listings
- L *n* for -P listings, specifies the optional hierarchy level (0-4) to be displayed. Default is the first tape level of the hierarchy.
- O print unordered "-l" or "-1" format listings
- P print one line per node with volume/position/VSN list/COS/file family/subsystem info
- R recursively list the directories
- T w|r|c|m select time field to display for long ("-l") listings w: last write (default) r: last read  
c: create m: last metadata update
- U print HPSS-specific information
- V print volume info for 1st tape level

- X print extended volume info (for all levels)
- 1 (numeral "one") forces one name-per-line list

## LSCONNECTIONS

SYNTAX:

**lscon[nections]**

ROLE:

Display active HPSS server connections

## LSCOS

SYNTAX:

**lscos** [-n] [-N] [ *drive...* ]

OPTIONS:

-n shows Named COS Lists available for the connection

-N *namedCosList*

list COS IDs associated with the specified Named COS list

*drive* one or more logical drive IDs, e.g. "lscos A: C:"

## LSFSET

See [LSFILES](#) (page 37).

## LSFILESETS

SYNTAX:

**lsfiles[ets]** [-l] [ *drive...* ] [ *filesetname* ]

ROLE:

Display Core Server filesets.

OPTIONS:

-l long listing

*drive* logical drive for the connection. Default is the current connection.

## LSJUNCTIONS

SYNTAX:

```
lsjun[ctions] [-s subsysID] [drive...]
```

ROLE:

Display Core Server junctions

-s Subsystem ID (default=1)

-s *drive* : logical drive for the connection. Default is the current connection.

## LSSITES

SYNTAX:

```
lssi[tes] [-a] [-c] [-h] [-m] [-p] [-s site[,site...]] [ drive...]
```

ROLE:

Display site entries from hsirc file(s)

-a list all info for sites

-c list server host name and connection info

-h list just server host info

-m list authentication method(s) defined for this site

-p list principal info for the site

-s list info for the specified site(s). The parameter may contain either a single site name, or a comma-separated list of sitemames. Sitemames are case sensitive.

drive optional space-separated list of drive letter(s) for which the site info is to be displayed, e.g.  
"a: z: "

## MDELETE

See [RM](#) (page 46).

## MD

See [MKDIR](#) (page 39).

## MDEL

See [DELETE](#) (page 25).

## MGET

See [GET](#) (page 30).

## MIGRATE

### SYNTAX:

```
mig[rate] [-h][- L level | -l level ][-F | -f][-P | -p][-R][-S] path...
```

### ROLE:

migrate files from disk cache to tape

- h if specified, treats symlinks as normal files. Default is to skip symlinks
- L or -l specifies hierarchy level from which to migrate data. The valid range is 0 (default) to 4
- F or -f force migration even if not needed
- R [standard option]recursively traverse directories in the specified path(s)
- P purge file from disk cache after migration
- S allow staging of file if no disk data.

## MKDIR

### SYNTAX:

```
mkd[ir] [-A " annotation " ] [-m mode ] [-p] path
```

### ROLE:

(a[dd], md, mkd[ir]) Make a new subdirectory node called path.

- A annotation string to set on newly created or preexisting directories. If -p is specified, and intermediate subdirectories are created, this option only applies to the final subdirectory in the path(s)
- m mode to use for creation of the directory(s). This must be an octal value in the range 0-7777. The mode value is silently truncated if it exceeds this range
- p Creates missing intermediate path name directories. If the -p flag is not specified, the parent directory of each newly-created directory must already exist.

## MOVE

See [MV](#) (page 40).

## MPUT

See [PUT](#) (page 44).

## MV

### SYNTAX:

```
mv [-f] [-i] [-v] path1 path2
```

### ROLE:

(move, ren[ame]) Move HPSS object *path1* to new HPSS object *path2*.

or

```
mv [-f] [-i] [-v] path1 path2... dirpath
```

Move HPSS objects into a directory.

### OPTIONS:

- f force option. Removes existing target if it already exists. Default is not to remove existing target(s)
- i in interactive mode, pause for confirmation before removing existing target

## NEWACCT

### SYNTAX:

```
newa[cct] [ accountID | accountName ]
```

### ROLE:

Set new account ID for current session or list available accounts if *accountID* (or *accountName*) is not specified.



## OPEN

### SYNTAX:

**open** [ *options* ]

### ROLE:

Open a new HPSS server connection Aliases: connect

### OPTIONS:

- A auth\_method
- c credentials cache filename (kerberos)
- d driveletter: (note ":" is optional)
- h remote host name or IP address
- k password-file (or keytab-file)
- l principal
- p portnumber
- q "quiet" mode (no Message of The Day) during startup
- s sitename

(A list of sitenames defined in the the hsirc or ~/.hsirc file can be obtained by using the LSSITE) command.

## OUT

### SYNTAX:

**out** [ - | [> | >|] | >> ] *localfile* ]

### ROLE:

Writes HSI listable output to localfile

### OPTIONS:

**out** closes any current output file and reverts to terminal output

**out file**, **out >file**, and **out >|file**

all attempt to open and write to a local file called file, overwriting any previous contents

**out >> file** attempts to append to an existing file

## PLOCK

### SYNTAX:

**plo[ck]** [-h] [-R] [-s] *path...*

### ROLE:

Lock files in HPSS disk cache

### OPTIONS:

- h read through symlinks as if they were regular files/directories [default: ignore symlinks]
- R recursively traverse directories in the specified path(s)
- s prestage files if necessary prior to locking This command should be used judiciously, as it may cause the disk cache to fill up with files that are not purgeable. Some sites may choose to disable this command - check with your site administrator to determine the local site policy for the use of this command

## PROGRESS

### SYNTAX:

**progress**

### ROLE:

Toggle display of file transfer progress.

## PROMPT

### SYNTAX:

**prompt**

Toggle for enabling or disabling interactive prompting for multiple file transfers (mget, mput, medelete commands).

## PUNL

### SYNTAX:

**punl[ock]** [-h] [-R] *path...*

### ROLE:

Unlock files the were previously locked in disk cache.

### OPTIONS:

- h read through symlinks as if they were regular files/directories [default: ignore symlinks]
- R recursively traverse directories in the specified path(s)

## PURGE

### SYNTAX:

**purge** [-F] [-R] [-h] [-l *hlevel* ] *path...*

### ROLE:

Purge files from HPSS disk cache at top level of a hierarchy

- F force purge to occur
- h if specified, treats symlinks as normal files. Default is to skip symlinks
- l purge data from hierarchy level *hlevel* (0-4) Default is level 0.
- R [standard option]recursively traverse directories in the specified path(s)

# PUT

## SYNTAX:

```
put [-A "string"] [-B | -b] [-d] [-F on | off ] [-h] [-M mode ] [-n]
      [-P | -p] [-Q] [-R | -r] [-T on | off] [-t] [-U | -u] path...
```

## OPTIONS:

- A annotation string for HPSS file (type "annotate ?" for more details)
- B | -b backup option. Renames existing HPSS file by appending "~"
- d remove local files after success transfer to HPSS
- F on | off enable or disable use of the HPSS Local File Mover, overriding any configuration settings in the HSIRC file(s)
- h symlink option. If specified, create symlink in HPSS if local symlink. If not set, reads through local symlinks when copying files to HPSS
- M *mode* specifies octal mode to use for file creations
- n *days* only put files modified within last "days" number of days
- P create intermediate HPSS subdirectories for the file(s) if they do not exist
- p preserve timestamp. Attempts to copy local file's timestamp to HPSS file
- Q if running as the HPSS "root" user, preserve the local owner/group for HPSS files and directories that are created
- R | -r recursively store directories
- T on | off enable or disable use of the HPSS Transfer Agent, overriding any configuration settings in the HSIRC file(s)
- t "re-put" operation. Restart a previously failed operation, using the size of the existing HPSS file as the starting offset.
- U | -u update option. Only copy file to HPSS if local file timestamp is newer

## PWD

### SYNTAX:

`pwd [ 0-9 ] [ drive :... ]`

### ROLE:

List current HPSS working directory path.

Working directory number (0 is the default if none is specified). `drive` : logical drive ID. Default is the current connection

## QUIT

### SYNTAX:

`q[uit]`

See END (page 27).

## R

### SYNTAX:

`r`

### ROLE:

Alias for "history -s". See HISTORY (page 32).

## RD

See RMDIR (page 46).

## REC

### SYNTAX:

`rec [v]`

See GET (page 30).

## REMOVE

See RMDIR (page 46).

## REPLACE

SYNTAX:

**rep[lace]**

or

**r[eplace]**

See [PUT](#) (page 44).

## RM

SYNTAX:

**rm** [-i] [-R] *path...*

ROLE:

(del[ete], mdel[ete], erase) Remove filename from HPSS.

OPTIONS:

-i                    Interactively confirm each delete.

-R                    Recursively remove files and subdirectories in the specified paths.

**Note:** MDELETE prompts for "yes/no" in interactive mode. Use the **prompt** **command** to toggle interactive mode. "-i" causes a prompt to be issued regardless of the "prompt" setting.

## RMDIR

SYNTAX:

**rmdir** *path...*

ROLE:

(rd, rem[ove]) Remove directory *path*. This command will fail if the directory is not empty

## SAVE

SYNTAX:

**s[ave]**

See also [CPUT](#) (page 24).

## SEND

See [PUT](#) (page 44).

## SET

### SYNTAX:

**se[t]**

### OPTIONS:

(defa[ult]) Set or display keyword values. These can be set for individual commands by either including them on the command line in the form "keyword=value", or for the entire session by using the command *set keyword=value*.

The **set** verb is optional; any line which is of the form *keyword = value* is interpreted as a *set* command.

You can also save a keyset (by using the *keep* command) and reload it later (by using the *adopt* command) in the same or another session.

ba[ckup]=on | off

automatically renames existing file on get/put. Default is on.

col[umns]= *numeric value*

Specifies number of columns per line. This value is used by the **ls** command when determining the number of entries per line to list. It is changed whenever the screen size changes.

copies=n

number of copies to store. The default is set by the HPSS administrator in the server configuration file.

cos[id]=auto | dualcopy | nodualcopy | *cosid*

class of services to use. Default is "auto", which selects class of service for you. You should only change this if you need to use a different class of service, for example, use "dualcopy" to go to two copies on tape for mission critical data. Use the *lscos* command to see the classes of service defined for the site; be aware that these are highly site-dependent.

**Note:** The **dualcopy** and **nodualcopy** settings are LLNL-specific options.

coslist= *name*

"named Class of Service" to use. Default is no named COS. Named Classes of Services are groups of COSs that are defined by the HPSS administrator. Use "*lscos -n*" to display a list of named COSs for the current active connection.

dcreate[mode]= *octal\_value*

Octal mode to use when creating directories. This value is defined in the global or private hsirc file's **dcreate\_mode** setting, or to 0777 by default. The current umask value is used in conjunction with the mode setting in order to yield the effective permissions that are set.

dirn= *pathname*

(n is omitted (meaning dir0) or 0 to 9) (The default pathname is your HPSS home directory) dir0 or dir assigns the current working directory. dirn sets up a shorthand name for the nth working directory as *pathname*. This directory can then be used in commands by using the "tic" notion. For example **ls 3**

family[id]= *numeric\_value*

Set default file family ID to be used when creating new files. The default is either 0, or the value defined in the global or private hsirc files for your site.

fcreate[mode]= *octal\_value*

Octal mode to use when creating files. This value is defined in the global or private hsirc file's **fcreate\_mode** setting, or to 0777 by default. The current umask value is used in conjunction with the mode setting in order to yield the effective permissions that are set.

iob[ufsize]= *numeric\_value*

I/O buffer size to use when transferring files. The default is set by the HPSS administrator when HSI is built, normally 8MB. The *numeric\_value* can optionally be followed by any of the following multipliers, with no intervening space: "k", "kb", "m", "mb", "g", "gb", "t", "tb", "p", "pb" for kilobytes/megabytes/gigabytes/terabytes/petabytes. The maximum buffersize is normally 32MB, but is defined at build time by the HPSS administrator.

lines= *numeric\_value*

Sets the number of lines per page for the terminal. This value was used for a curses-based help package for a previous version of HSI, but is currently unused.

promptl[en]= *numeric\_value*

Sets the prompt length in characters

promptd[irlen]= *numeric\_value*

Sets the maximum number of characters to use when expanding the %w metacharacter in the command prompt. This is the number of directory components to include from the righthand side of the pathname if the prompt must be truncated to *promptlen* characters.

PS1= *string*

Specifies the prompt string to use for initial command prompting. See the [Customizing the HSI Prompt section](#) (page 7) for details on the options for *string*. The default setting for this value is "?" or the value of the PS1 setting in the global or private hsirc file.



PS2= *string*

Specifies the prompt string to use for continuation lines. See the [Customizing the HSI Prompt section](#) (page 7) for details on the options for string. The default setting for this value is "> " or the value of the PS2 setting in the global or private hsirc file.

pwid[th]= *auto / numeric\_value*

Sets the parallel stripe width to use when transferring files. If **pwidth** is set to "auto" (the normal case), HSI automatically determines the optimum stripe width to use.

You can set these keywords for individual commands, for your session, or in saved keysets.

## SETCON

SYNTAX:

```
setc[on] conid
```

ROLE:

Set current active HPSS server connection.

OPTIONS:

*conid*            numeric handle to current connection, as shown by the **lscon** command

## SETDRIVE

SYNTAX:

```
setdr[ive] connection|drive newdrive
```

ROLE:

Change the "logical drive" letter for an HPSS connection

OPTIONS:

*connection*        connection handle ID (1- *n* ), as shown by the **lsconnections** command

*drive*              current drive letter (a-z, A-Z)

*newdrive*          drive letter to be assigned (a-z, A-Z)

## SHARED\_MEM

SYNTAX:

**shared\_mem**

ROLE:

On/off toggle for enabling/disabling the use of shared memory for I/O buffers for file transfers.

## SHOW

SYNTAX:

**sh[ow]** [-l] [-h] [keyset...]

ROLE:

Display names of kept keysets, or if

**keyset**

is specified, shows its contents.

OPTIONS:

-l                lists local keysets.

-h                (default) lists HPSS keysets.

## SHOWCON

SYNTAX:

**showc[on]**

See LSCONNECTIONS (page 37).

## STAGE

### SYNTAX:

**sta[ge]** [-A] [-i] [-m max] [-N] [-R] [-w] [path...] [<<MARKER]

### ROLE:

Stage file(s) from tape to disk cache within an HPSS hierarchy

### OPTIONS:

- A default) enable tape mount autoscheduling optimization
- i no interactive notification when file(s) have been staged
- m max number of stages to issue in a single batch. Default=no limit
- N disable autoscheduling optimization of stage requests
- R recursively traverse directories in the specified path(s)
- w wait for each stage to complete (foreground stage). Default is to stage in the background

**MARKER** provides sh-style "here"-document syntax for specifying filenames. Lines following the **MARKER**, which must be the last token on the line, contain one or more pathnames, (which may optionally include wildcards). The list is terminated by a line containing **MARKER** as the first token on the line.

## STORE

### SYNTAX:

**st[ore]**

See [PUT](#) (page 44).

## SU

### SYNTAX:

```
su [- | -l | -m] [ user | #uid ]
```

### ROLE:

Assume identity of another HPSS user.

### OPTIONS:

- same as "-l"
- l full login with target user's environment (deferred implementation)
- m login with current user's environment (default)
- user* login name of the user whose identity is to be assumed
- #uid* numeric user id (the "#" prefix is required, with no intervening whitespace) of the user whose identity is to be assumed. Default is root (uid 0)

## SUDO

### SYNTAX:

```
sudo [-s] [-u user|#uid] [--] command
```

### ROLE:

Issue command(s) as another HPSS user sudo [K | -k | -l] [-v]

### OPTIONS:

- K "sure kill" option - invalidate the current sudo session and remove the starting timestamp
- k invalidate the current sudo session, but do not remove the starting timestamp
- l (deferred implementation) list allowed and forbidden commands for the current active sudo session
- s SU to the target user and then run the command. Stays in "SU" mode after the command completes
- u target user login name or numeric uid (# prefix is required, with no intervening whitespace)
- v validate and extend the current SUDO session's timestamp
- command* HSI command to be executed.

## SWITCH

SYNTAX:

**swi**[**tch**]

See [SETCON](#) (page 49).

## TOUCH

SYNTAX:

**t**[**ouch**] [-c *cos ID*] [-M *mode* ] [-R] *hpsspath...*

ROLE:

Touch HPSS file *hpsspath*. Creates HPSS file *hpsspath* if it does not exist or updates the time last referenced if it exists.

OPTIONS:

- c specifies the Class of Service ID to use when creating new files. The `lscos` command can be used to view a list of available COS IDs.
- M specifies the octal mode bits to set when creating a new file, for example: -M 0662 to set read/write for owner and group.
- R recursively traverse directories in the specified path(s)

## UMASK

SYNTAX:

**umask** *permission-mask*

ROLE:

Sets the permission mask to the octal number specified by *permission-mask*. (See `man umask` for more information about `umask`.)

## VERBOSE

SYNTAX:

**verbose**

ROLE:

Toggle VERBOSE mode.

## VERSION

SYNTAX:

`vers[ion]`

ROLE:

Show HSI version information.

## WHOAMI

SYNTAX:

`whoami`

ROLE:

Display current HPSS user identity

# Multiple HPSS Systems

This section contains information on the following topics:

- Using HSI with multiple HPSS systems that are linked together via Junctions and Cross Cell authentication
- Using HSI to access multiple HPSS systems
- Opening connections to more than one HPSS system during a single HSI session
- Description of the *Logical Drive* pathname syntax that HSI uses to associate files with a connection during a session
- List of commands that are used to open, close, and switch between active sessions
- Examples of storing, retrieving, and copying files while connected to multiple HPSS systems

## Using HSI to Access Multiple HPSS Systems

HSI has the ability to establish concurrent connections to more than one HPSS system at more than one site (or to the same HPSS site) within a single session. After the sessions are opened, normal HSI commands can be used to work with any of the sessions, as described below. Files can be copied between the sites using one of the following mechanisms:

- Third-party transfers, which do not require the data to flow through HSI memory
- First party transfers, in cases where firewalls or other network connectivity problems prevent the movers from the source HPSS system from connecting to the HSI Gateway Server at the sink HPSS system.

The command that is used to copy files between sites is described below.

There are three requirements that must be met in order to use the Multi-HPSS feature:

- You must have an account on each of the HPSS systems that you wish to access
- There must be network connectivity between the client system on which HSI is executing, and the HPSS site(s) that you wish to work with.
- In order to transfer files using the 3rd-party mechanism, there must be network connectivity between the mover nodes on the source HPSS system, and the HSI Gateway server process on the sink HPSS system.

When working with HPSS systems, HSI treats each connection as a "logical drive," borrowing from the familiar legacy of the early personal computers. In this environment, there are three key concepts:

1. The notion of *current connection*.

This is initially established when the first connection is opened as HSI begins executing. It can be changed during execution as connections to other HPSS systems are opened and as you switch between logical drives. (See below.) Each connection has its own separate context, containing items such as:

- The address of your HPSS server host
- The logical drive letter that you (or HSI) assigned for the connection
- Your home current directory
- The current working directory
- The classes of service that are available on the HPSS system

2. The use of *logical drive* notation as a command prefix and to reference files and directories on a particular HPSS system.

Logical drive notation simply consists of a non-case-sensitive *drive letter* followed by the colon (:) character (with no intervening whitespace). For example:

**D:** followed by the command or pathname with no intervening spaces. For example:

**d:ls** would switch to connection D: and list the files in the current working directory

**Ls C:\*.\*** Lists all files ending with the characters ".c" on the HPSS system associated with logical drive C:.

**get fromHPSS1 : D:file1 fromHPSS2 : F:file2**

Fetches the file *file1* from the HPSS system associated with drive D:, renaming it to the local file "fromHPSS1" and similarly would fetch the file *file2* from the HPSS associated with drive F:, renaming it locally to "fromHPSS2"

**Note:** As of HSI version 2.4, the colon character (:) used to separate the local and HPSS pathnames must be surrounded by whitespace. This requirement was added in order to avoid ambiguity in the inclusion of colon characters in pathnames. You can tell which HSI version you are using by typing:

**version**

Drive letters are not case sensitive, so the following are equivalent

**f:get myfile**

**F:get myfile**

It is not necessary to use this notation; commands and pathnames without a drive prefix always reference the current connection. If the multi-HPSS feature is not used, the drive letter syntax can be disregarded with a few exceptions. Depending upon the context, the use of a standalone drive letter followed by a colon, e.g. "b:" or "ls C:", may be interpreted as either of the following:

Set the current connection to the HPSS system associated with the logical drive (first example, above)

or

Reference the current working directory for the HPSS system associated with the logical drive (second example, above)

In commands where a filename is required, the use of a standalone drive letter will cause a usage error, for example:

**get P:**

### 3. Third Party copies

When copying files between HPSS systems, the default is to use a 3rd-party copy mechanism so that the data flows directly between HPSS systems, without having HSI in the middle of the transfer. See the CP command in Chapter 8 for more details. For some situations, such as copying files through a firewall, it may be necessary to use a store-and-forward mechanism to read data from either a local file or from an HPSS system inside the firewall, and then transfer it to an HPSS system outside the firewall. See the firewall command for more information on setting up store-and-forward transfers.



## Commands Used for Multi-HPSS HSI Sessions

Commands used to access multiple HPSS systems are shown in the table below, and are also described in more detail in the [HSI Command Directory](#) (page 17). The minimum abbreviation for each command is shown in the text preceding the "[" character.

open or connect

Establishes a new connection to an HPSS system.

close

Closes an existing connection to an HPSS system.

lscon[nections] or showc[on]

Shows a list of currently established connections, including the home directory and the current working directory for the connection.

lssi[tes]

Displays a list of site names that have been defined in the global .hsirc startup file, or in your private .hsirc file.

*drive:*

As a standalone command, this changes the current active connection.

*drive:command*

Changes the current active connection, then issues the specified command in the context of the new connection.

setdrive

Changes the logical drive letter assigned to a connection.

## Example Scenario for a Multi-HPSS Session

A typical interactive scenario for using HSI to communicate with multiple HPSS systems is described below.

Set the prompt string so that we can tell which logical drive is our current active connection and what our current working directory is within that connection.

We want our prompt string to look like this:

```
drive_letter:[hostname]working_directory->
```

To accomplish this, the PS1 variable can be set in the hsirc file, or interactively as follows:

```
PS1 = "%d[%H]%w3->"
```

which will yield a prompt string similar to this:

```
C:[hpss]/home/gleicher->
```

The %w3 will cause long working directory pathnames to be displayed by truncating to at most three subdirectories on the right-hand part of the prompt string, and replacing characters in the middle of the pathname with the string "...", e.g.

```
S:[hpss05i]/users.../tools/scripts/backman->pwd
```

```
pwd0: /users/sdsc/gleicher/HPSS/tools/scripts/backman
```

## Establish Initial Connection

If the site administrator has set up the global `hsirc` file, or you have customized your private `.hsirc` file in your home directory, then all that is required is to startup HSI with no options (to connect to the site-defined default HPSS system), or

```
hsi -s sitename
```

to connect to the site called *sitename*. Note that this is a case-sensitive option, which must exactly match the sitename contained in the `hsirc` stanza.

If the `-s` option is not used on the command line, the following optional parameters may be used to connect to any HPSS system that you have an account on:

```
hsi -h hostname -p port -l login_name -A authentication method  
-k keytab filename
```

The format of the login name varies, depending on the authentication method that you have chosen. For the `combo` method (`-A combo`) and the `keytab` method (`-A keytab`), the login name is your principal name on the HPSS system to which you are connecting. At most sites, this will be the same as your login name on any other machines at that site, but check with the HPSS administrator at the site to make sure.

For `kerberos` authentication (`-A kerberos`), the login name is normally your principal name and the Kerberos Realm name, which is generally in uppercase, e.g.:

```
-A kerberos -l betty@UNM.EDU
```

(See "The HSI Startup File" for more information about the contents of the `HSIRC` file)

Normally, the site administrator will set up an HSI wrapper script containing all the information regarding hostnames, ports, and authentication mechanism(s) to be used, so that the optional parameters shown below are usually not needed.

## Establish Connection to Another Site

As above, if the site administrator has set up a global `hsirc` file, or if you have customized your private `.hsirc` file to add the appropriate information, the command:

```
open -s new_sitename
```

or

```
open -d drive_letter:
```

can be used to establish a connection to the HPSS system designated by *new\_sitename*. or *drive\_letter*.

The `lssites` (page 38) command can be useful when running interactively, if you need help remembering the exact sitename(s) and/or drive letter(s) that have been set up by the HPSS administrators. For example, at Gleicher Enterprises, the following sites are defined:

```
R:[hcdp01]/users/sdsc/gleicher->lssites
```

```
Site, Logical Drive....gel.62, "G:"
```

```
Site, Logical Drive....ornl.probe, "P:"
```

```
Site, Logical Drive....ornl.prod, "O:"
```

```

Site, Logical Drive....nersc.prod, "N:"
Site, Logical Drive....sdsc.prod, "R:"
Site, Logical Drive....sdsc.test, "T:"
Site, Logical Drive....iu.prod, "I:"

```

**Note:** In this example, I am currently logged into HPSS at SDSC and am going to open a connection to Indiana University:

```

R:[hcdp01]/users/sdsc/gleicher->cd HPSS/tools/scripts/backman
R:[hcdp01]/users/sdsc/gleicher->open -d I: -A combo
Password: password
I:[hpss]/home/gleicher->

```

Display the list of active sites after opening the second connection:

```
R:[hpss]/home/gleicher->lscon
```

```

List of Currently Active Connections
Current default connection handle: 3

```

Drive	Handle	Remote IP Address	Remote Hostname	HSM Level	Fire-wall	I/O Mode	Ctl Port	Data Min	Port Max	Cmd Count
R:	1	132.249.20.148	hcdp01.sdsc.edu	6.2.1	off	normal	1217	0	65535	22
HomeDir: /users/sdsc/gleicher										
pwd0: /users/sdsc/gleicher/HPSS/tools/scripts/backman										
->I:	3	131.215.145.148	hpssbl.uits.indiana.edu	6.2.1	off	normal	1218	0	0	13
HomeDir: /home/gleicher										
pwd0: /home/gleicher										

```
C:[hpss]/home/gleicher->
```

At this point, standard HSI commands can be used to list directories, store, retrieve and copy files using the logical drive pathname syntax, etc.

In the next example, we want to store a copy of the local file "good\_stuff" on one of the HPSS systems, and then replicate it on the other, in order to provide disaster recovery. We will do this by using the cp command, so that the data flows directly between the two HPSS systems, without having HSI in the middle of the transfer.

First, store the local file on logical drive R: in the directory big\_project, which is an existing subdirectory directly underneath our home directory on both HPSS systems:

```

R:[ hpss]/home/gleicher->put good_stuff ~/big_project/good_stuff
put good_stuff : /home/gleicher/big_project/good_stuff
(1083916 bytes, 397.5 KBS (cos=6))

```

Next, copy the file from the R: drive to the I: drive:

```

R:[hpss]/home/gleicher->cp big_project/good_stuff i:~/temp/good_stuff
cp R:/home/gleicher/big_project/good_stuff to

```

```
I:/home/gleicher/temp/good_stuff
```

```
(2001/09/16 23:36:18 1083916 bytes, 481.1 KBS (cos=12))
```

**Note:** The "cp R: ..." on the 2nd and 3rd lines are the HSI command response.

At this point, we are ready to close out the connection(s) and terminate the session(s). This can be done by simply exiting out of HSI. If we want to continue to work in one of the connections, but have no need to keep the connection open for the the other, then the close command can be used, e.g.:

```
R:[hpss]/home/gleicher->close R:
```

```
I:[hpss05i]/home/gleicher->
```

Notice that when the current active connection is closed, HSI switches to one of the other active connections, in this case, the only other active connection was drive I:. If an attempt is made to close this connection, the following error message will be displayed:

```
I:[hpss05i]/home/gleicher/temp->close i:
```

```
*** Cannot use <close> for the only open connection.
```

```
Use end (or one of its aliases) to terminate HSI
```

## Logging HSI Sessions

When HSI execution has terminated, a logfile containing a time-stamped record of all commands will be present in the hsilog file, which is normally written in the home directory. The default setting is not to write a logfile.

This can be controlled by using the HSI "LOG" command, or by setting the HSI\_LOGFILE environment variable as follows:

If using CSH:

```
setenv HSI_LOGFILE somepath
```

If using KSH

```
export HSI_LOGFILE=somepath
```

If using SH (this will also work for KSH and BASH)

```
export HSI_LOGFILE HSI_LOGFILE=somepath
```

*somepath* can be the reserved string **none** to disable automatic creation of a logfile. It can also contain metacharacters of the form %X as follows:

- %H : is expanded to your local home directory pathname
- %N : is expanded to the non-canonical local hostname, i.e., the hostname without any domain name. "snuffles.sdsc.edu" would become "snuffles".
- %P : is expanded to the current HSI process ID
- %U : is expanded to the local login name.

The default setting is usually determined by the HPSS administrator, and set in the HSI wrapper script. If the path already exists when HSI attempts to open the local logfile, HSI will try to append to it.

# HSI Environment Variables

HSI and its support libraries use several environment variables to control the program behavior. The method of setting these variables depends on the shell that is in effect:

For *csh*, environmental variables are set by using the syntax:

```
setenv environment_variable=value
```

For *ssh*, *ksh* or *bash*, environmental variables are set by using the syntax:

```
setenv environment_variable=value
```

For *csh*, environmental variables are set by using the syntax:

```
export environment_variable  
environment_variable=value
```

The following environmental variables are defined for the HSI package:

## **HPSS\_USE\_NETRC\_FILE**

If this variable is defined, HSI will attempt to read a `.netrc` file to obtain the password to be used for either the *local* (**-A local** command-line option) or *combo* (**-A combo** command-line option) authentication method.

## **HPSS\_SERVER\_HOST**

Defines the HSI Gateway Server host or IP address. This is normally defined by the HPSS administrator.

## **HPSS\_PRINCIPAL**

Defines the default principal name to use when logging into HPSS. The usual default is your local login name, or the name specified by the **-l** (ell) command-line parameter.

## **HPSS\_HOSTNAME**

Defines the default network interface that HSI should use when communicating with HPSS movers. This can be overridden by the HPSS administrator via a configuration file.

## **HSI\_HISTSIZE**

Defines the number of commands that are saved for use by the HSI **history** command

## **HPSS\_AUTHMETHOD**

Defines the default authentication mechanism to use. The mechanism must be one that was defined for both client and HSI Gateway Server when HSI was installed.

## **HPSS\_KEYTAB\_PATH**

If the “keytab” authentication method is used, this defines the pathname to the user's keytab file. This can be overridden by the HPSS administrator via a configuration file, or via the user's local `.hsirc` file, or via the **-k** command line option.

## **HPSS\_API\_DEBUG**

Defines the HSI library debugging level. These levels are bit settings, roughly corresponding to the logging levels defined for “syslog”:

- 2^0 : [LOG\_EMERG] system is unusable
- 2^1 : [LOG\_ALERT] action must be taken immediately
- 2^2 : [LOG\_CRIT] critical conditions
- 2^3 : [LOG\_ERR] error conditions
- 2^4 : [LOG\_WARNING] warning conditions
- 2^5 : [LOG\_NOTICE] normal but significant condition
- 2^6 : [LOG\_INFO] informational
- 2^7 : [LOG\_DEBUG] debug-level messages

## **HPSS\_API\_DEBUG\_FLAGS**

Defines the types of information that are logged. The string consists of a set of one-character flags as follows:

- w** - if set, the log file is truncated to zero length or created for writing
- a** - (default) if set, the log file is created if it does not exist or positioned to the end of the file, and all writes occur at the end of the file.
- f** - if set, causes the log file to be flushed after each message is written.
- p** - if set, causes pid to be included in message prefix.
- r** - if set, cause realtime (hi-res) time to be included in message prefix
- R** - if set, disable hi-res time in message prefix.
- t** - (default) if set, causes time from ctime(time()) to be included in msg prefix.
- T** - if set, disables **t** flag.

## **KRB\_SERVICE**

Defines the service name to be used by kerberos authentication. The default is defined by the HPSS administrator when HSI is installed.

## **KRB5CCNAME**

Defines the Kerberos credentials cache filename, if kerberos authentication is being used.

# Disclaimer

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

---

# Keyword Index

To see an alphabetical list of keywords for this document, consult the next section (page 65).

Keyword	Description
<u>entire</u>	This entire document.
<u>title</u>	The name of this document.
<u>scope</u>	Topics covered in this document.
<u>availability</u>	Where these programs run.
<u>who</u>	Who to contact for assistance.
<u>introduction</u>	Background terms and distinctions.
<u>authentication</u>	Authentication methods.
<u>how to run hsi</u>	Diagram and analysis of LC firewall.
<u>hsi help</u>	HSI help information.
<u>hsi command line options</u>	Basic HSI functionality.
<u>filenames</u>	Correct case for HPSS and Unix filenames.
<u>pathnames</u>	Pathname syntax.
<u>ftp-compatibility</u>	FTP compatibility.
<u>operating system considerations</u>	OS-specific issues.
<u>hsi-commands</u>	Instructions for firewall-altered services.
<u>customizing the hsi prompt</u>	Customizes the HSI prompt.
<u>hsi-to-access-multiple-hpss</u>	Access multiple HPSS systems
<u>hsi with multiple hpss</u>	HSI with Multiple HPSS Systems
<u>example multi-hpss sessions</u>	Examples of multi-HPSS sessions
<u>establish initial connection</u>	Establishing initial connection
<u>connecting to other sites</u>	Connecting to other sites
<u>logging hsi sessions</u>	Logging HSI sessions
<u>hsi environment variables</u>	HSI environmental variables
<u>examples</u>	Sample commands.
<u>customization</u>	HPSS Gateway machine settings.
<u>bugs</u>	Bug list.
<u>index</u>	The structural index of keywords.
<u>a</u>	The alphabetical index of keywords.
<u>date</u>	The latest changes to this document.
<u>revisions</u>	The complete revision history.



# Alphabetical List of Keywords

Keyword	Description
<u>a</u>	The alphabetical index of keywords.
<u>authentication</u>	Authentication methods
<u>availability</u>	Where these programs run.
<u>hsi command line options</u>	Basic HSI functionality.
<u>bugs</u>	Bug list.
<u>commands for multi-hpss-sessions</u>	Commands for Multi-HPSS Sessions
<u>connecting to other sites</u>	Connecting to other sites
<u>customization</u>	HPSS Gateway machine settings.
<u>date</u>	The latest changes to this document.
<u>entire</u>	This entire document.
<u>establish initial connection</u>	Establishing initial connection
<u>examples</u>	Sample commands.
<u>example multi-hpss sessions</u>	Examples of multi-HPSS sessions
<u>filenames</u>	Correct case for HPSS and Unix filenames.
<u>ftp-compatibility</u>	FTP compatibility.
<u>how-to-run-hsi</u>	Explains how to run HSI.
<u>hsi command line options</u>	Basic HSI functionality.
<u>hsi-commands</u>	Instructions for firewall-altered services.
<u>hsi environment variables</u>	HSI environmental variables
<u>hsi-help</u>	HSI help information.
<u>hsi to access multiple hpss</u>	Access multiple HPSS systems
<u>hsi with multiple hpss</u>	HSI with Multiple HPSS Systems
<u>index</u>	The structural index of keywords.
<u>introduction</u>	Background terms and distinctions.
<u>logging hsi sessions</u>	Logging HSI sessions
<u>operating system considerations</u>	OS-specific issues.
<u>pathnames</u>	Pathname syntax.
<u>customizing the hsi prompt</u>	Prompt string settings.
<u>revisions</u>	The complete revision history.
<u>scope</u>	Topics covered in this document.
<u>title</u>	The name of this document.
<u>who</u>	Who to contact for assistance.

## Date and Revisions

Revision Date -----	Keyword Affected -----	Description of Change -----
05Oct08	entire	First edition of HSI manual.

ANG (13Oct08)

UCRL-WEB-201524

LLNL Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

ANG (22Oct08) Contact on the OCF: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov), on the SCF: [lc-hotline@pop.llnl.gov](mailto:lc-hotline@pop.llnl.gov)